The World is Too Big to Download: 3D Model Retrieval for World-Scale Augmented Reality

Yi-Zhen (Angela) Tsai, James Luo, Yunshu Wang, and Jiasi Chen
University of California, Riverside

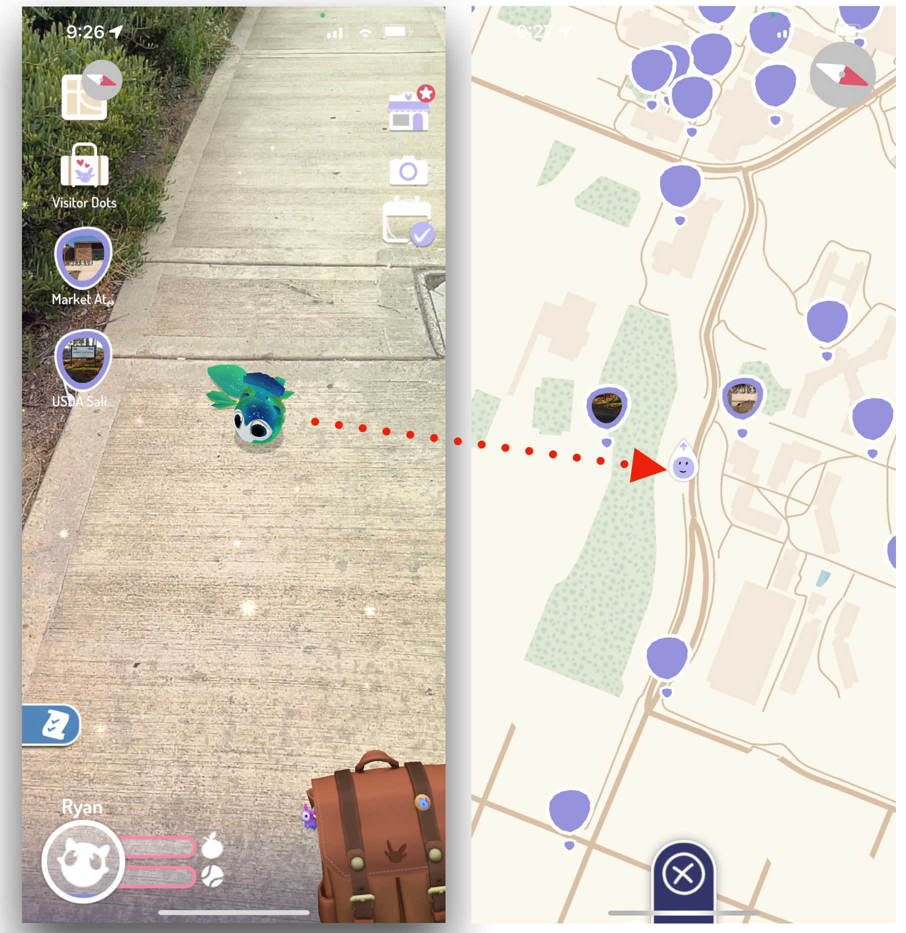
UC RIVERSIDE

World-Scale Augmented Reality

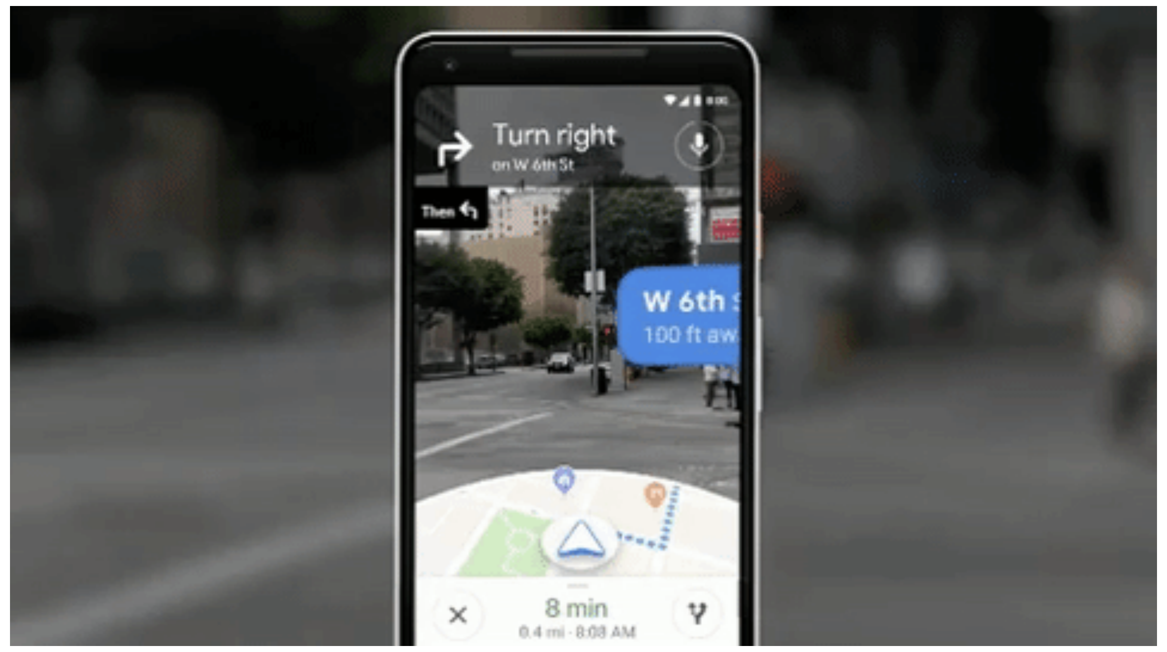
- World-scale AR uses your **physical** location, and overlays digital content around you

- Envision larger scale of augmented reality applications with 3D models around (Game creatures, Advertisements, virtual navigation signs, etc.)

Pokemon Go



Peridot by Niantic



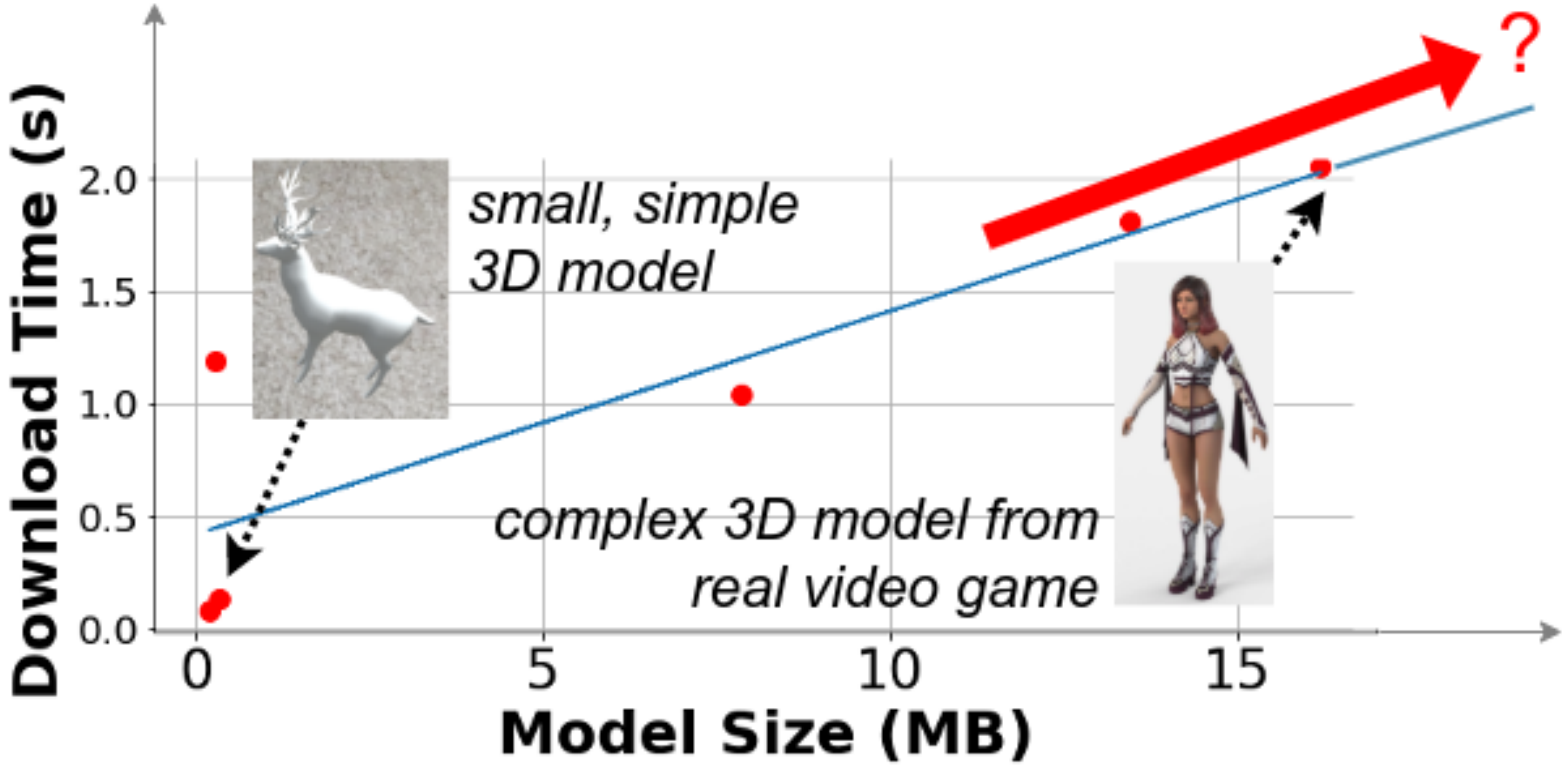
Google Maps Live View

World-Scale AR

UC RIVERSIDE

Challenges for World-scale AR

- Given geographical scale, it’s impossible to pre-fetch all 3D models beforehand
- As 3D models become more complex, the file size and the retrieval time increase



1 model = 2 seconds, 15 MB

100 models?
200 seconds, 1.5 GB!

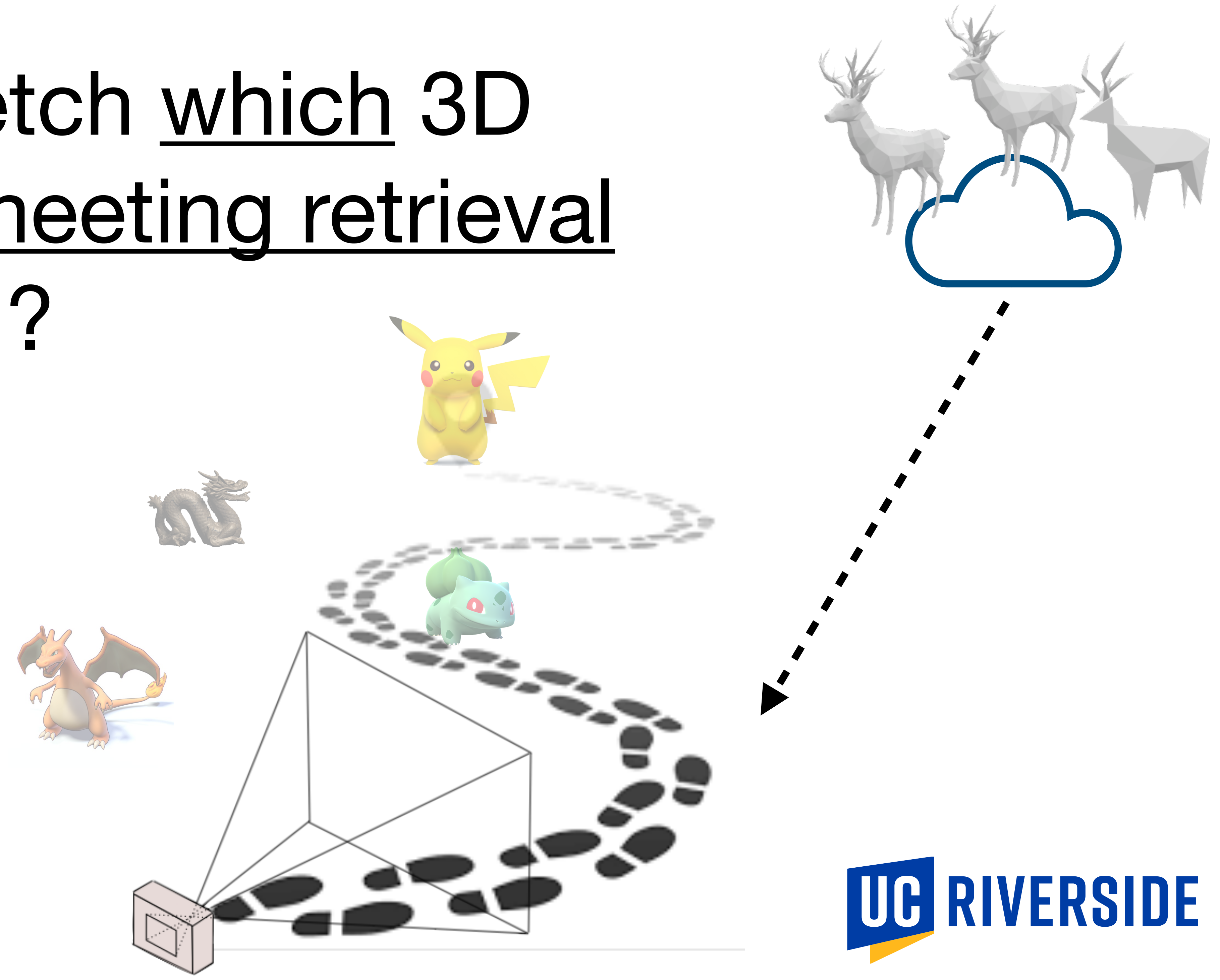
Only download what is needed on-the-fly

UC RIVERSIDE

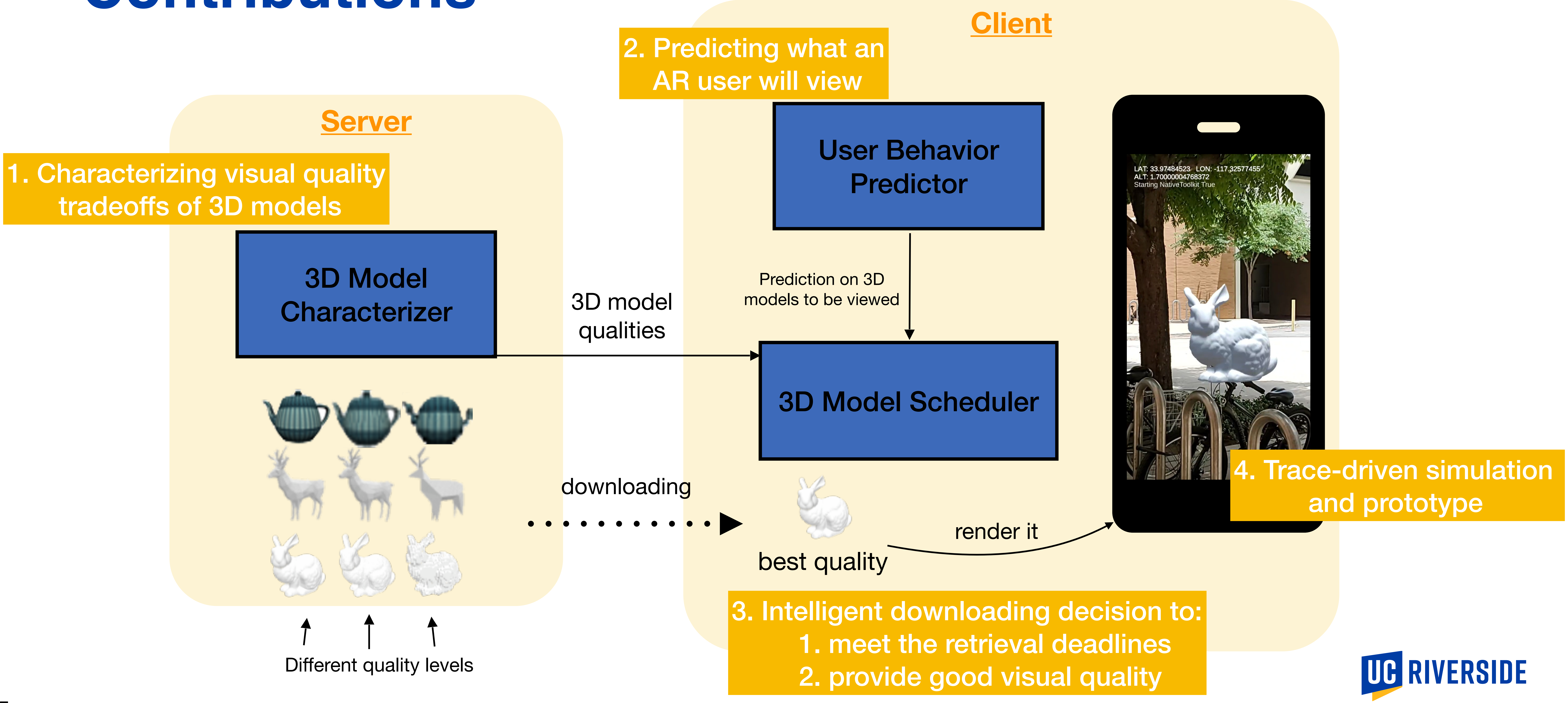
Problem Statement

- **Problem**: In the world-scale AR, with complex 3D models at specific locations around the world, users want to view them quickly and with high visual quality

- **Research question**: <u>When</u> should the app fetch <u>which</u> 3D models to <u>maximize the visual quality</u> while <u>meeting retrieval deadline</u> as the user moves around the world?



Contributions



1. 3D Model Scheduler (client)

- Optimization Goal:
  - Maximize the visual quality (utility) of the selected version for each 3D model
  - Meet the retrieval deadline of each 3D model within available network bandwidth
- Decide:
  - Selected 3D models at specific quality level
  - Retrieval order

UC RIVERSIDE

Formal Optimization Problem

Objective

$$\max \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{r=1}^{N} \overbrace{u_{ij}}x_{\underbrace{ijr}}$$

Visual quality, derived from 3D Model Characterizer

Make the decision to maximize visual quality

Decision variable of whether to download 3D model i at version j in the r[th] place

Size of 3D selected model

Subject to

$$s.t.\ \overbrace{s_{ij}}x_{ijr} \leq \sum_{t'=t-d_{ijt}}^{t} B_{t'}\ \forall i,j,r,t$$

Selected models can be downloaded in time (meet retrieval deadlines)

$$D_r = \sum_{r'=1}^{r} \sum_{i=1}^{N} \sum_{j=1}^{M} d_{ijD_{r'}} x_{ijr'} \leq \sum_{i=1}^{N} \sum_{j=1}^{M} \underbrace{t_i} x_{ijr}\ \forall r$$

Deadline, derived from User Behavior Prediction Module

$$\sum_{j=1}^{M} \sum_{r=1}^{N} x_{ijr} = 1\ \forall i$$

Only one version of each 3D model should be selected

$$\sum_{i=1}^{N} \sum_{j=1}^{M} x_{ijr} = 1\ \forall r$$

Only one 3D model is downloaded at a time

$$variables\ x_{ijr} \in \{0,1\}, d_{ijt} \in N$$

→ Item selection problem

Earliest deadline first scheduling → DP can solve!

UC RIVERSIDE

Strategies of retrieving 3D models

Distance-based:

Retrieve all models within a radius r of the user

Models downloaded: 10

Unused: 7

Missed: 1

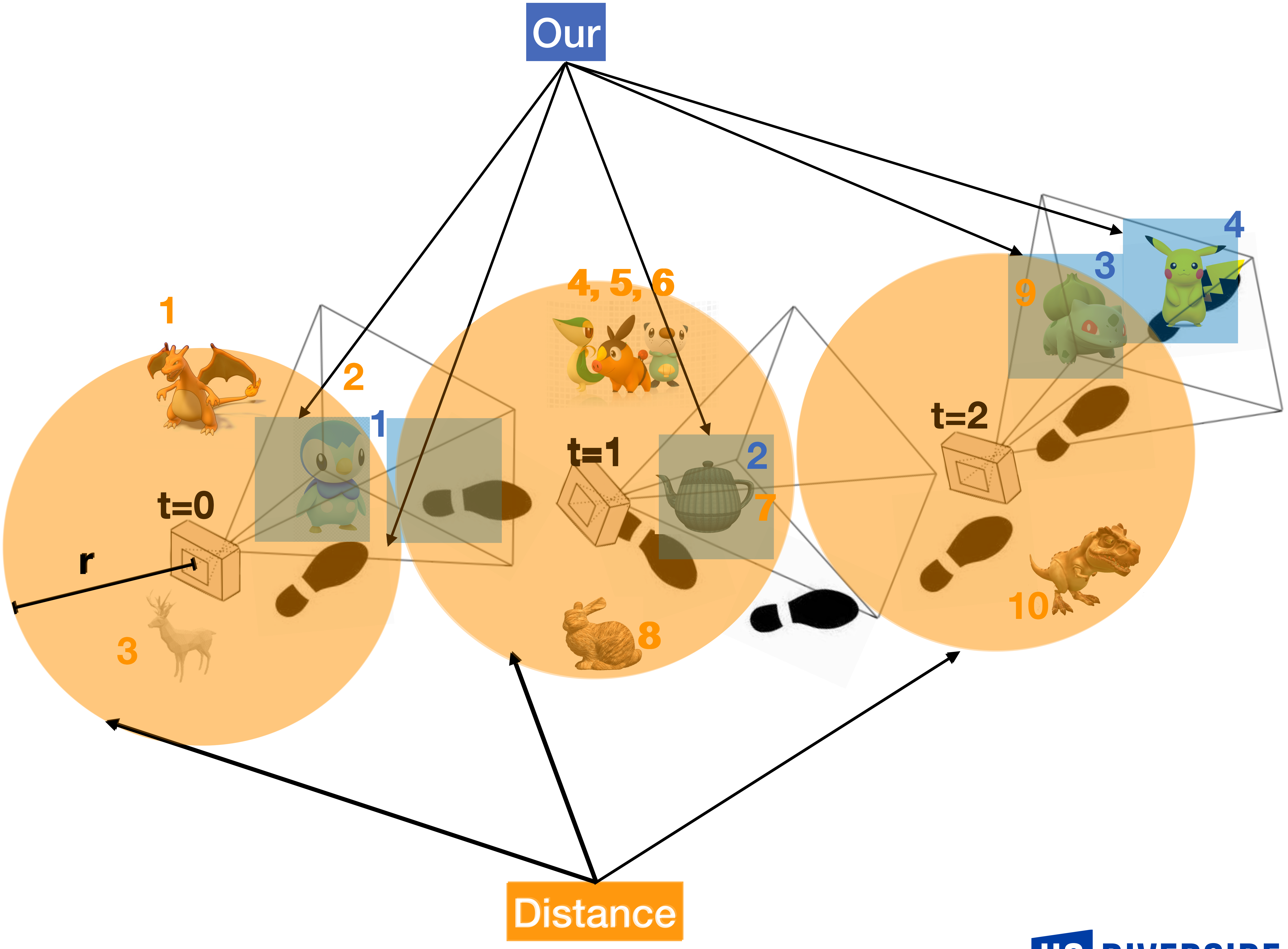
Our:

Retrieve models in the cells predicted by our User Behavior Predictor module

Models downloaded: 4

Unused: 0

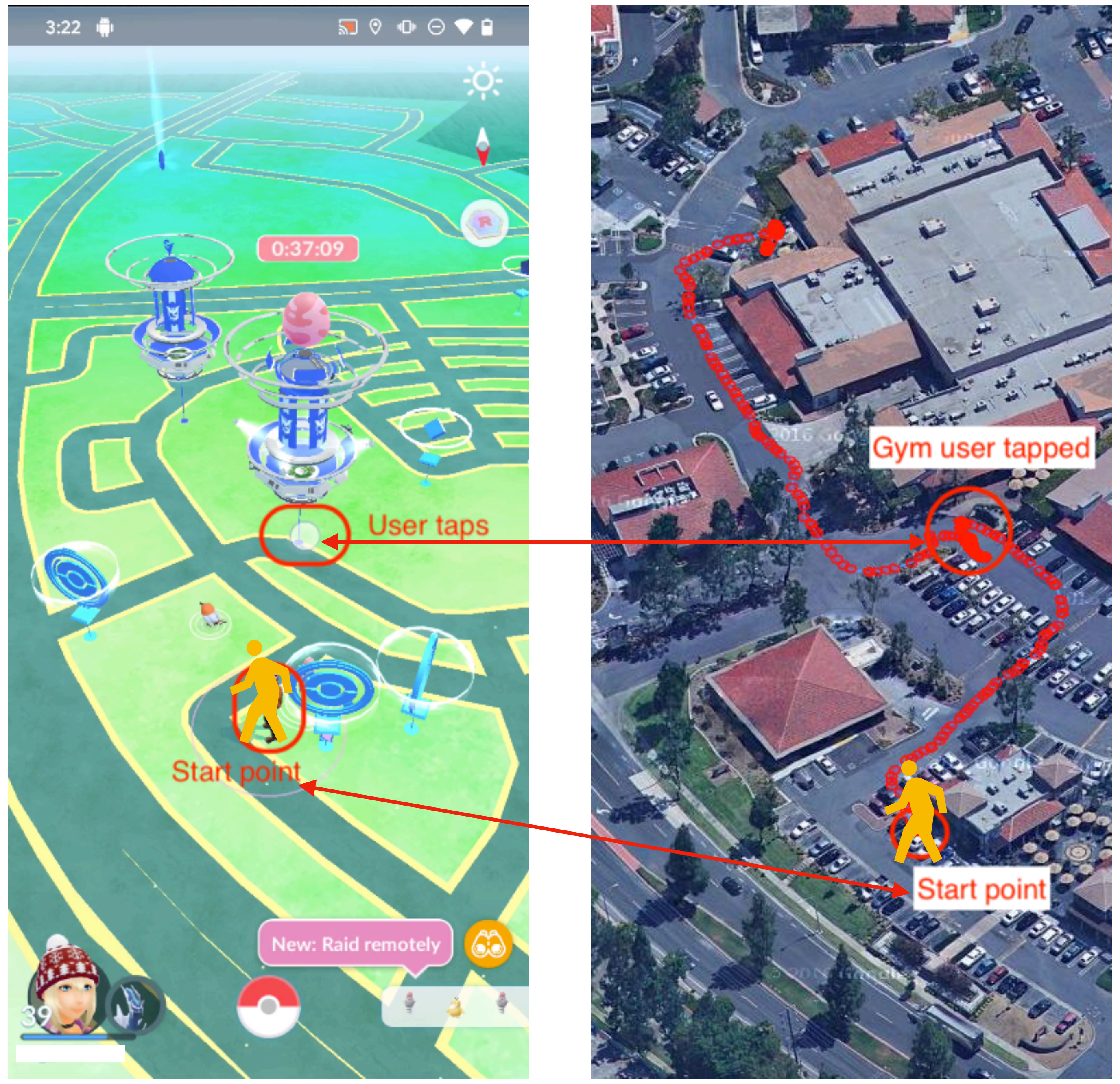
Missed: 0



2. User Behavior Predictor (client)

<u>Goal:</u>

Predict which 3D models user is likely to view next and their retrieval deadlines



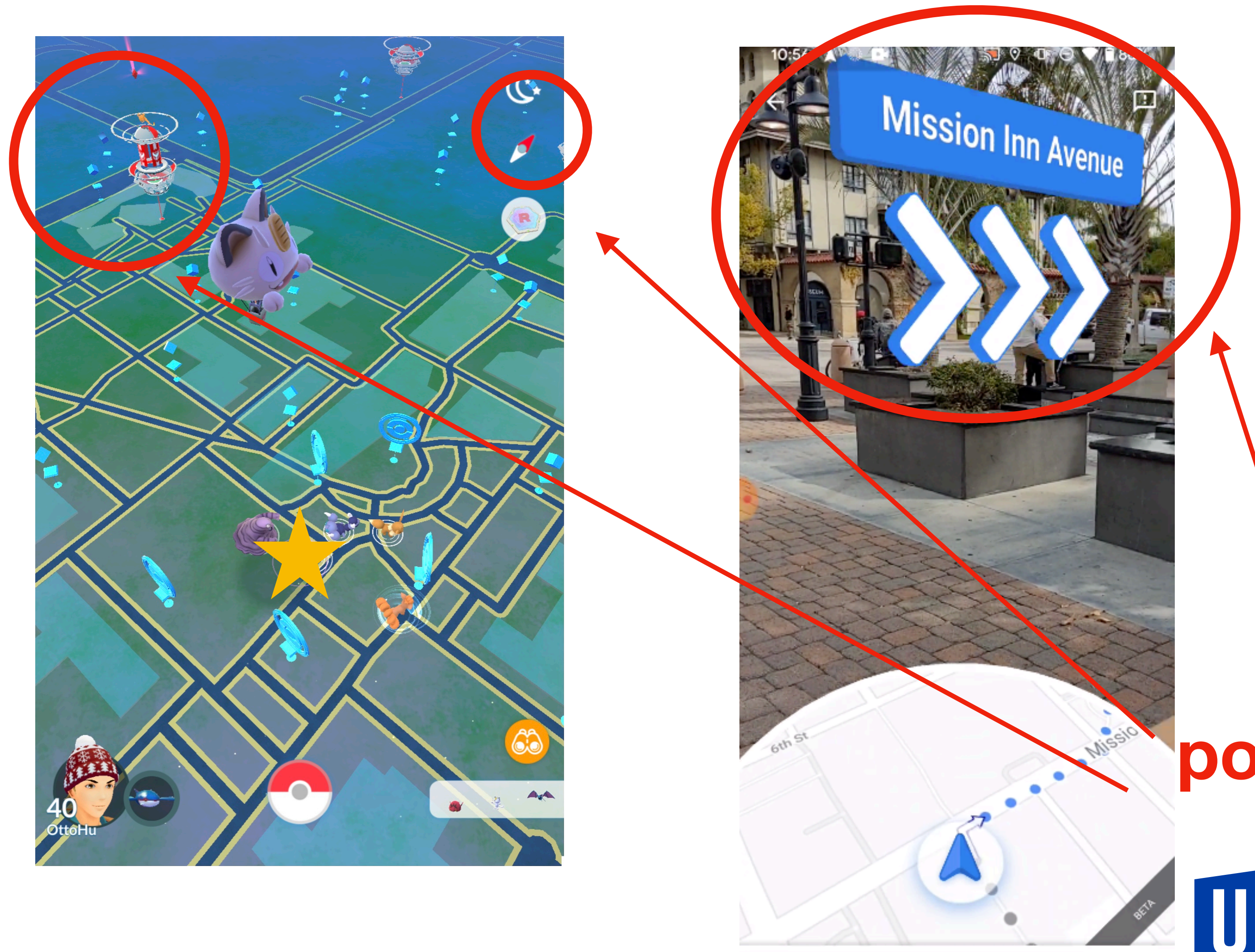
AR app screenshot

Corresponding trajectory on the satellite map

<u>To drive the predictor,</u>

we conduct a user study to measure the volunteers' behavior when playing Pokemon Go

- Geolocation data: history of user's movement from GPS/IMU
- AR display: location of 3D models on AR display, application features, user gestures



points of interests

Screenshots from AR apps

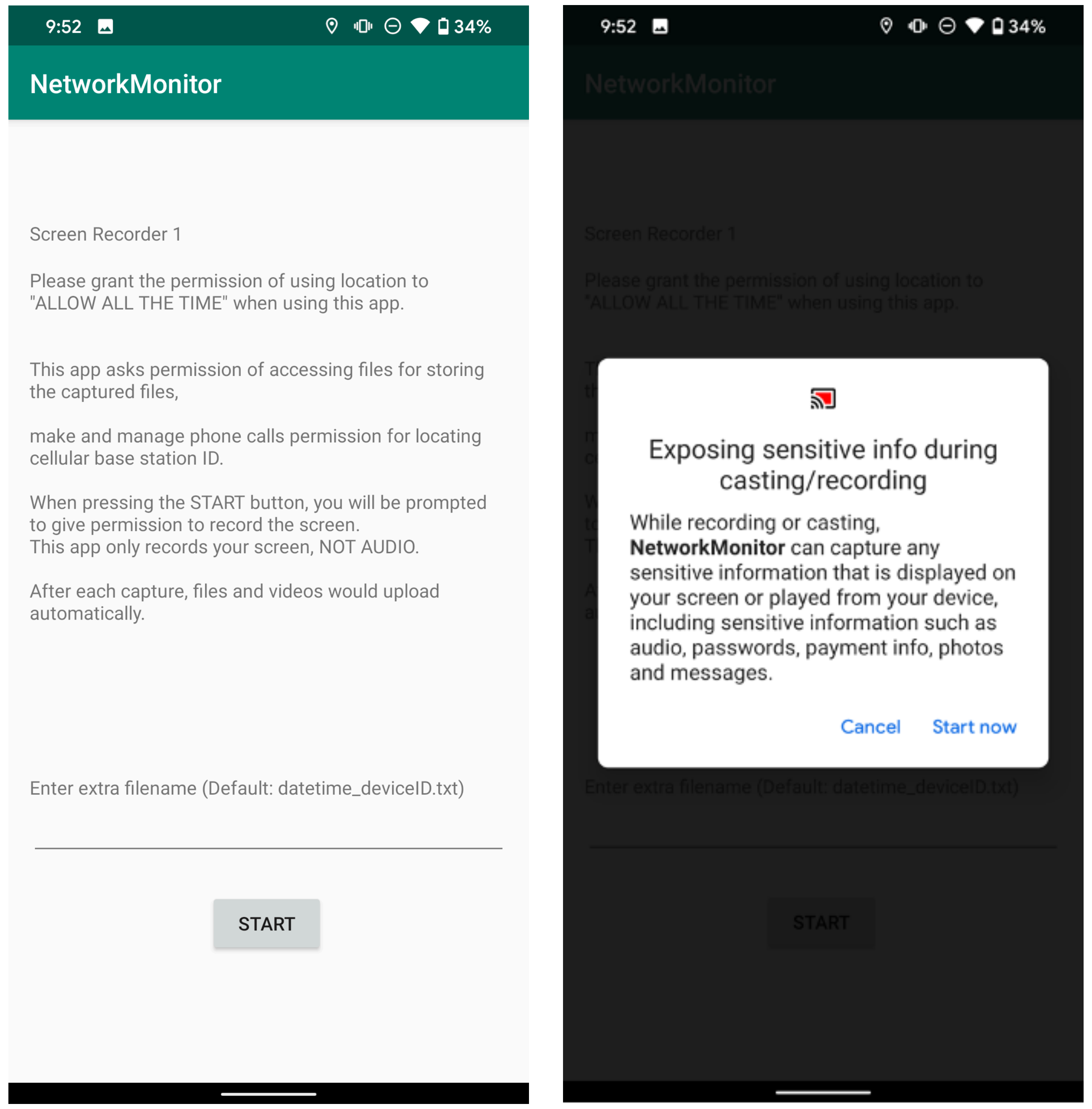
UC RIVERSIDE

<u>Hypothesis:</u>

<u>AR display provides hints of where user is likely to go next</u>

IRB-approved user study

- 7 volunteers from 6 different zones (parks, outdoor malls, university campuses) across multiple US states for several weeks
- 289 minutes of Pokemon Go and corresponding GPS/IMU traces are collected and analyzed



UC RIVERSIDE

Design of User Behavior Predictor (client)

How to generalize the predictor across multiple geographic zones?

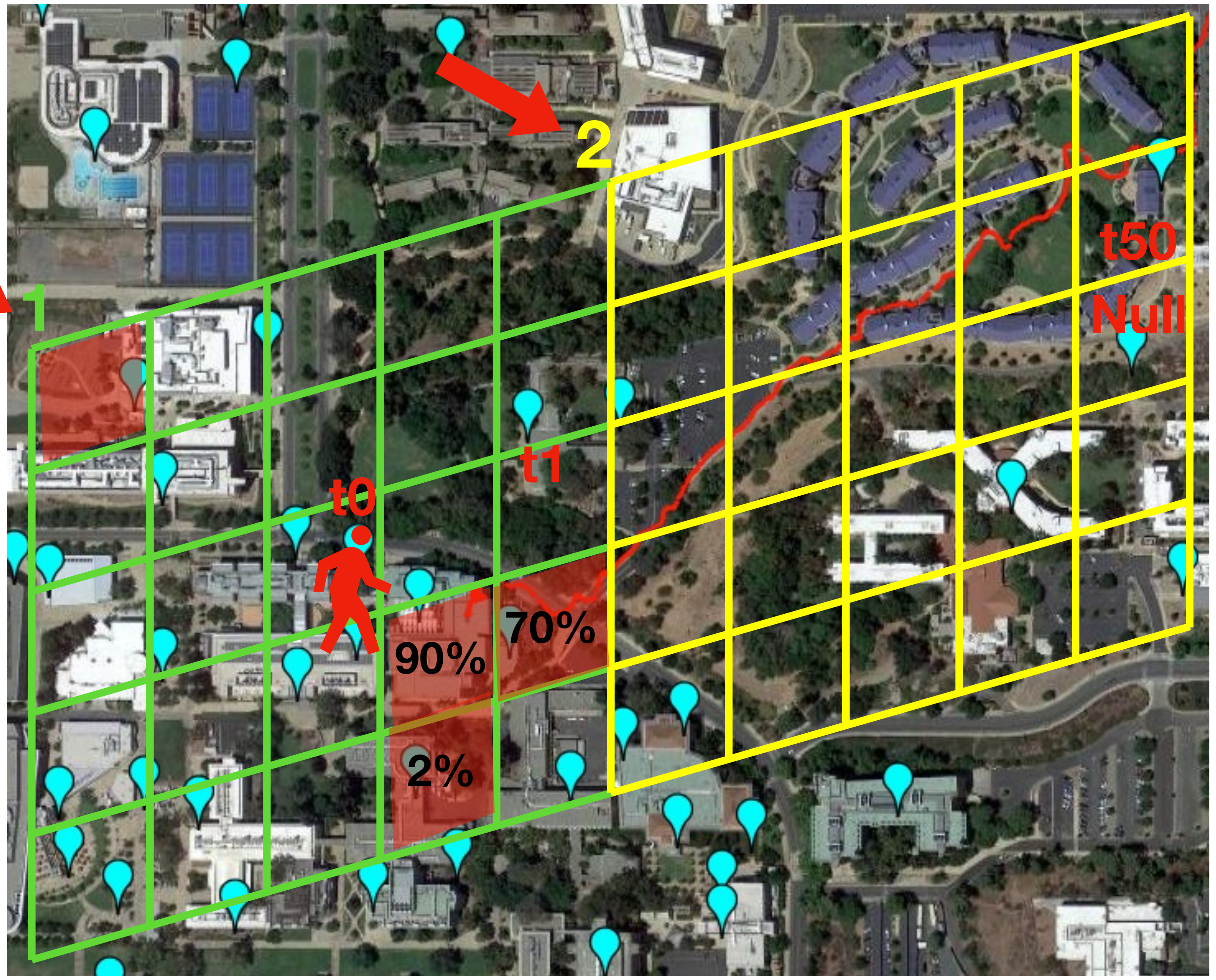
Convert absolute coordinates into sub-zones and cells [1], predict the cells instead

How many 3D models will the user visit in the near future?

Set confidence threshold for the predictor result to let the predictor decides

How far into the future to predict?

“Null” class to prevent predicting too far into the future

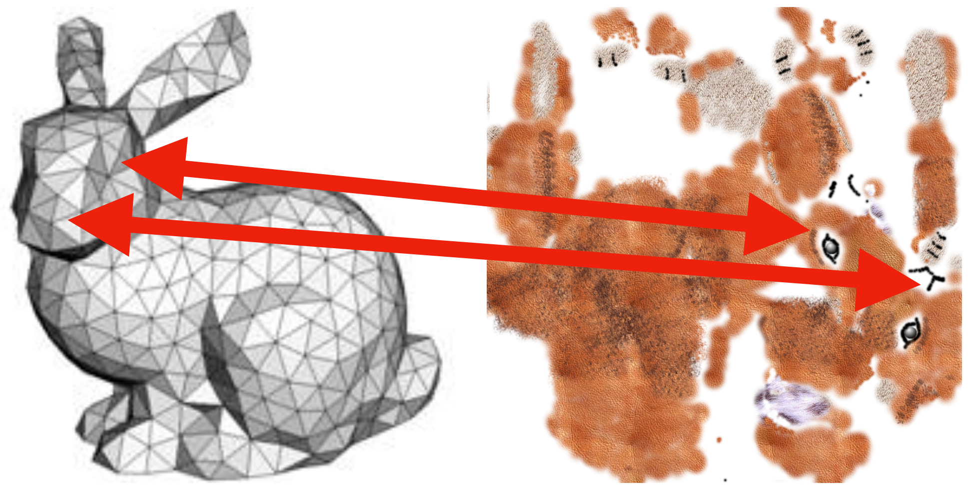


Given the past H seconds of feature data, predicts the 3D models that the user will view next, up to T seconds ahead.

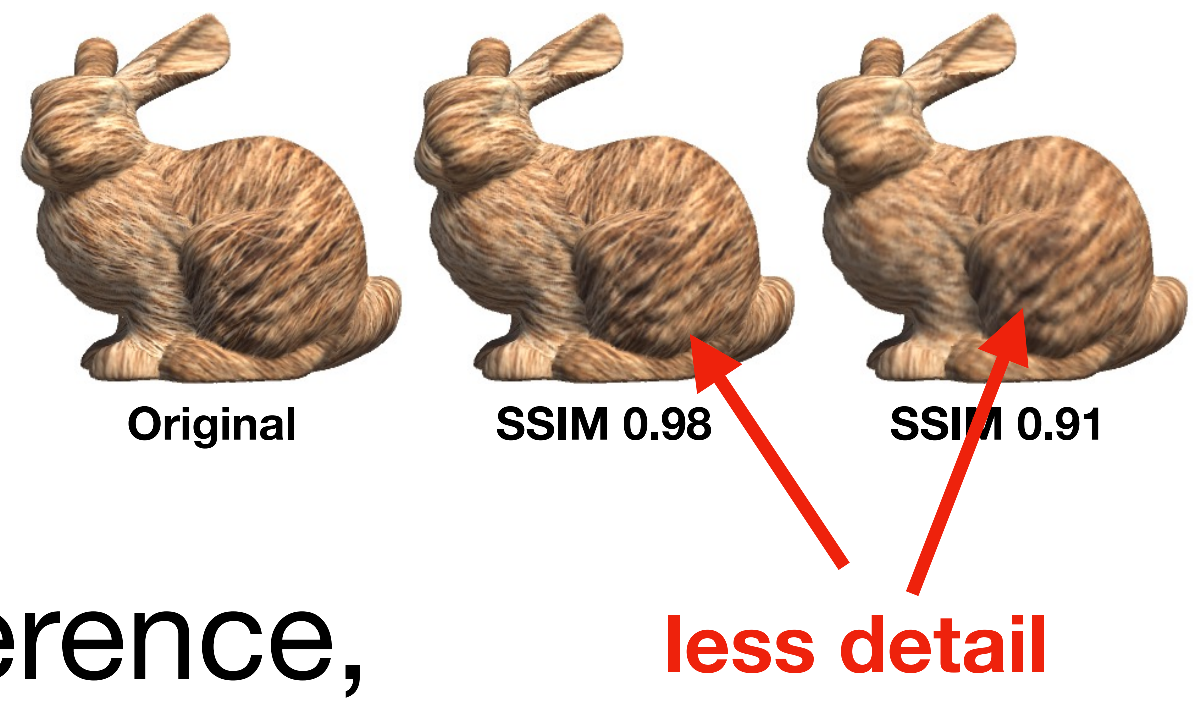
[1] s2 cell. https://s2geometry.io/devguide/s2cell_hierarchy.html

UC RIVERSIDE

3. 3D Model Characterizer (server)



- Problem:
  - Find out the if there is **trade-off between visual quality and latency**
- Approach:
  - **Estimate the visual quality** of each **compressed version** of 3D model
- 3D model compression parameters:
  - Mesh quantization: The geometry of a 3D model (shape)
  - Texture quality: Flat images applies to 3D model (skin)



- Metric: Structural similarity index measure (SSIM): a full reference, perception-based image quality metric, range from 0 to 1[2]

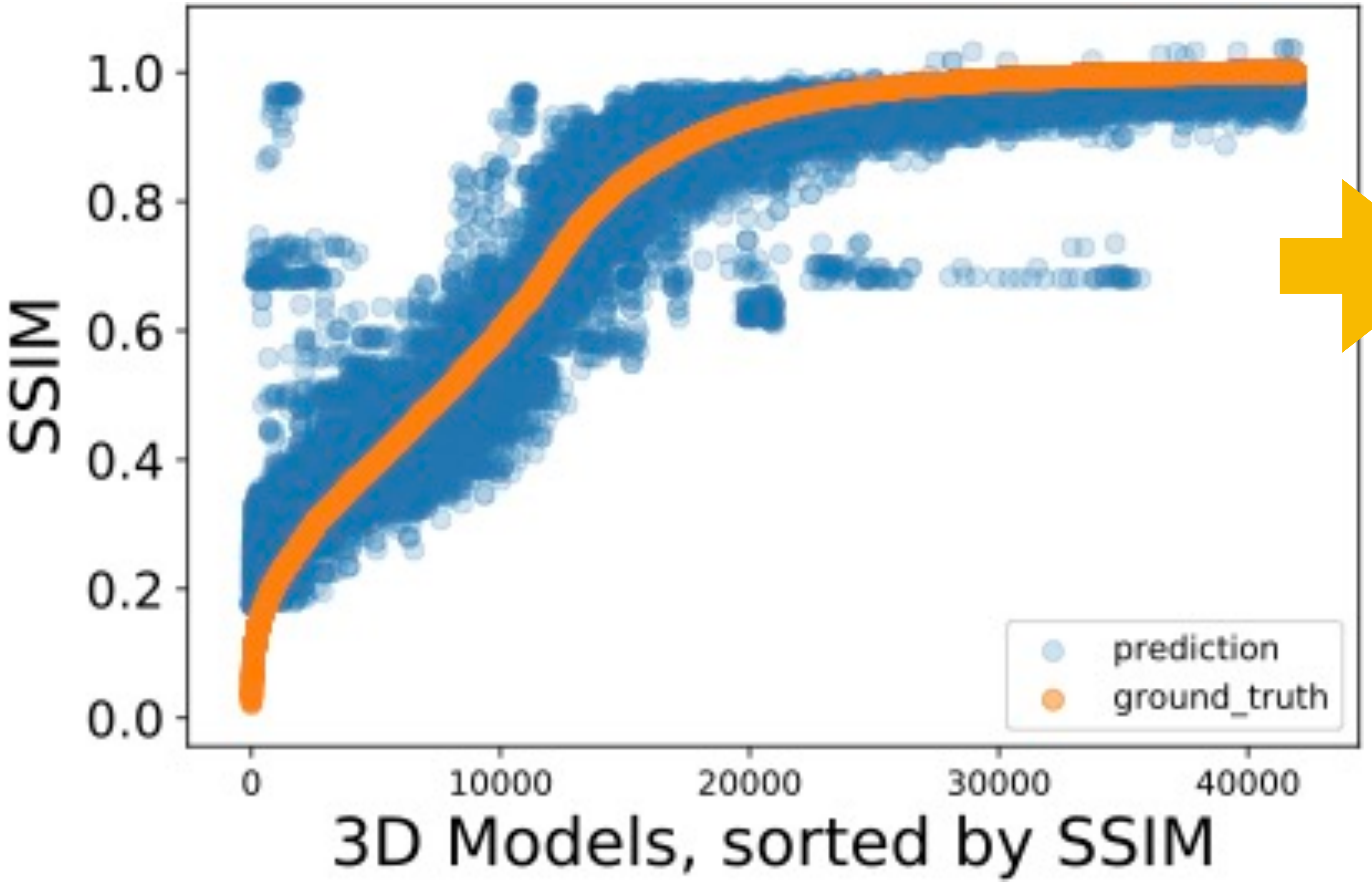
[1] figures from https://www.creatis.insa-lyon.fr/site7/en/acvd
[2] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity,", IEEE TIP '04

UC RIVERSIDE

Evaluation

Evaluation of 3D Model Characterizer (server)

- Data collection:14 popular base 3D models, 5 texture files to do data augmentation to create 40,000 models with different compression parameters
- Pearson's correlation coef. indicates that mesh and texture correlate with SSIM the most



SSIM prediction result

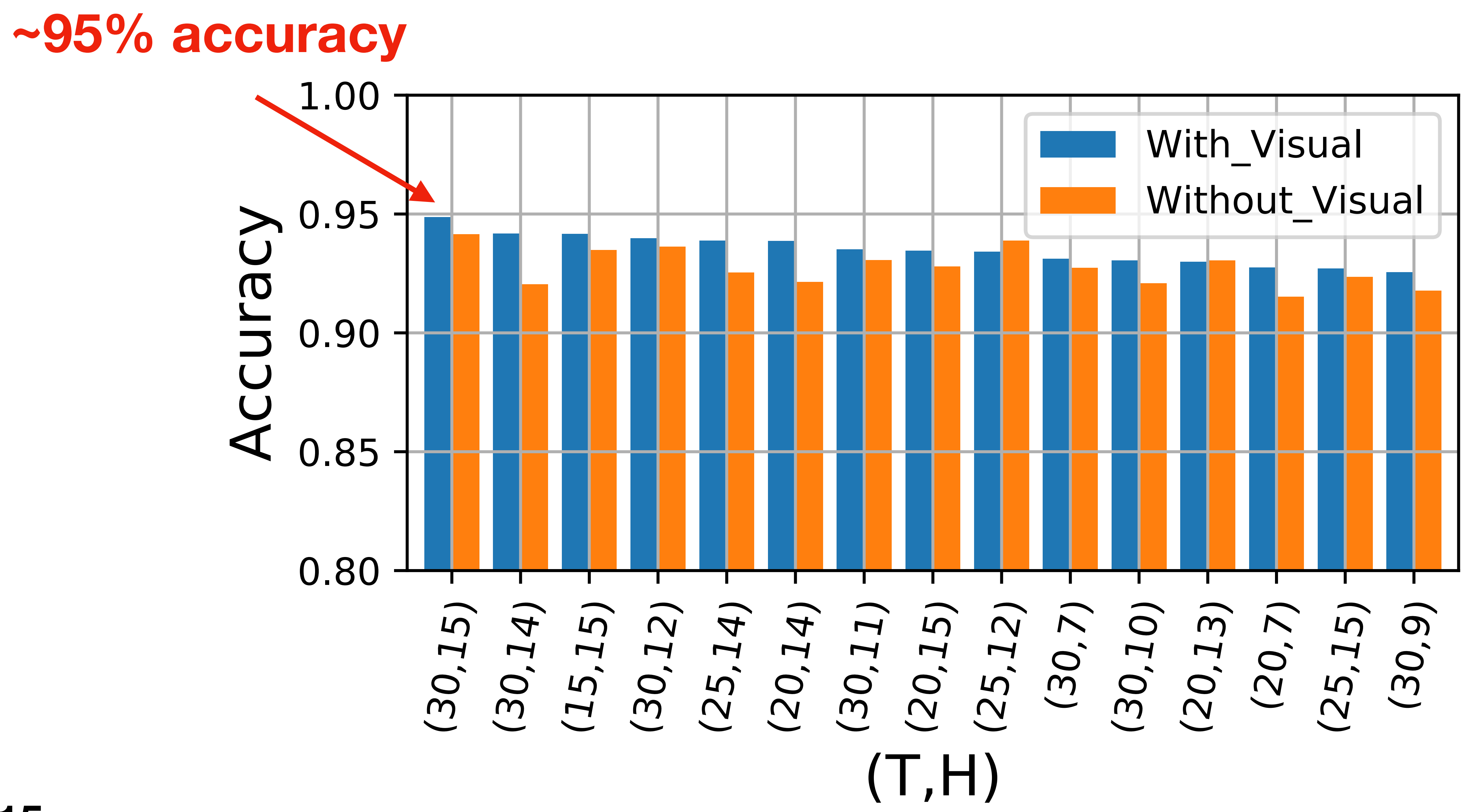
Our prediction results shows a good performance with an average error of 0.04 and Pearson's correlation coef. of 0.968

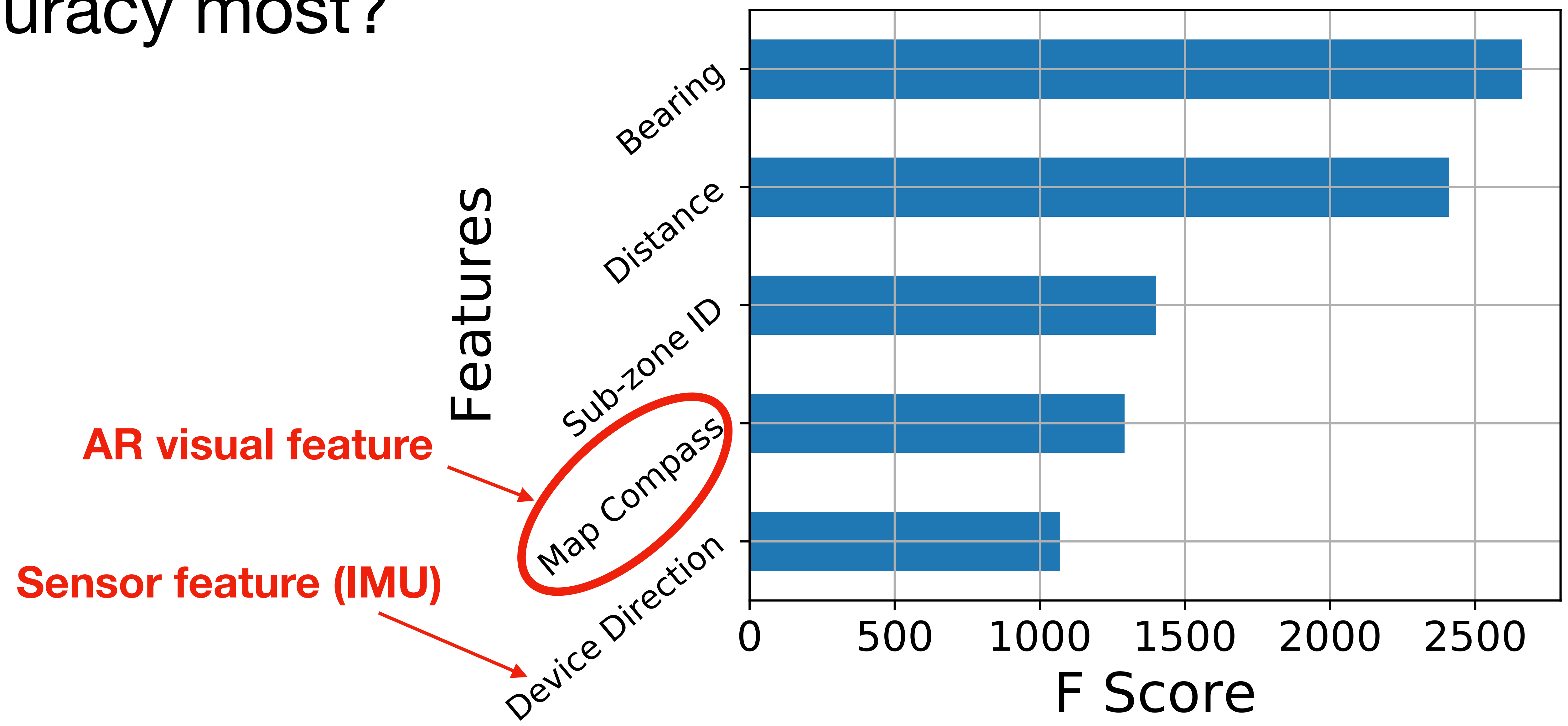
	Overall	Mesh < 8	Mesh >= 8
Mesh	0.775	**0.693**	0.176
Texture	0.023	-0.006	**0.221**

Correlation Coefficient between features and visual quality

Evaluation of the User Behavior Predictor

- Prediction model: Gradient boosted decision tree model to perform multi-class classification with 25 possible cells and the "null" class
- Our prediction accuracy is larger than 91%, with the help of AR Visual data, it can achieve up to ~95%
- Which feature helps the prediction accuracy most?





UC RIVERSIDE

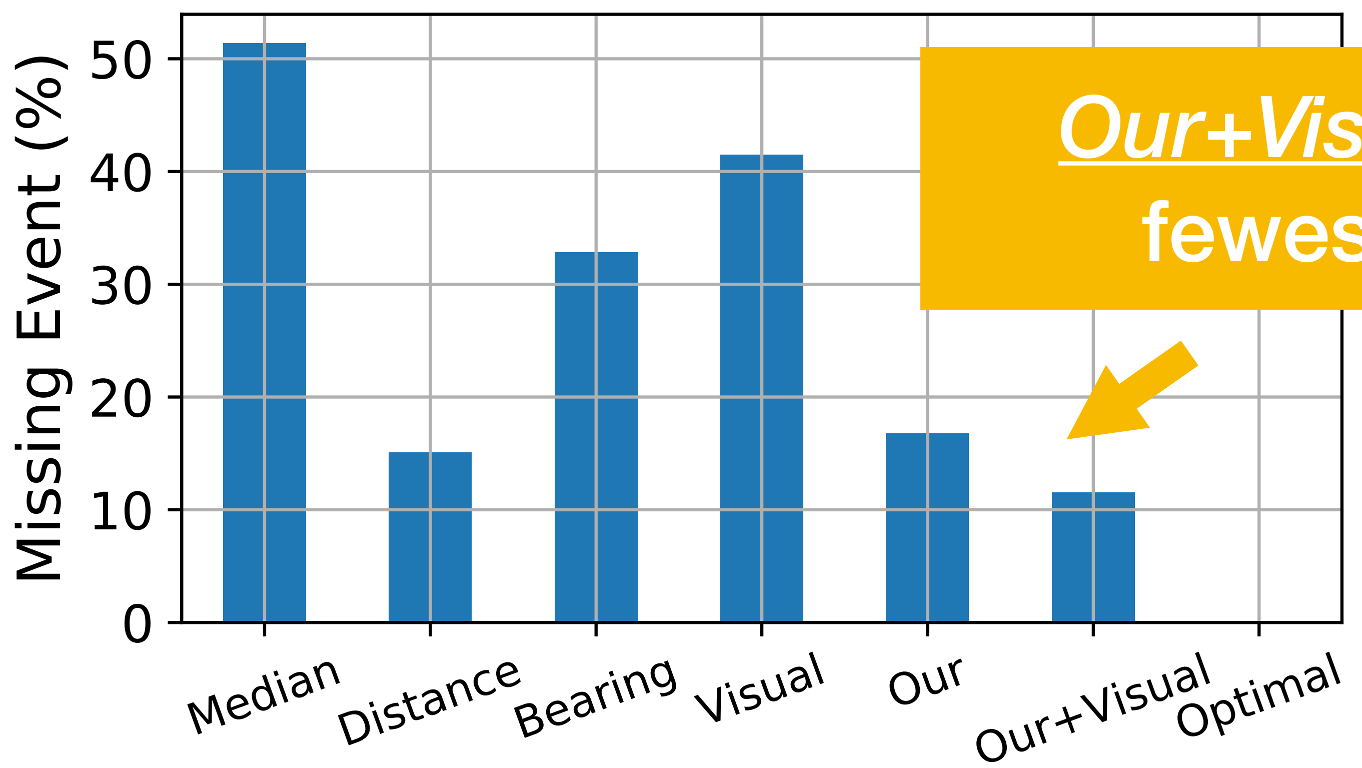
End-to-end Simulation

- Setup:
  - Real world-scale AR application **user traces in 4 zones**, with physical sensor data and AR visual data
  - **LRU cache** of size 100MB (half of total size of all median-quality 3D models in one zone)
  - **Variable outdoor 5G network bandwidth** sampled from Lumos5g dataset[1]
- Our+Visual method: Only those 3D models that are predicted to be viewed by the User Behavior Predictor are sorted and fetched by the 3D Model Scheduler
- Baselines:
  - *Median:* Always retrieve median quality of 3D models in random order
  - *Distance:* Retrieve 3D models within the circular range
  - *Bearing:* uses GPS coordinates' bearing information as prediction of future cells
  - *Visual:* uses only AR visual data "Map Compass" as prediction of future cells
  - *Our:* uses simplified User Behavior Predictor without AR visual data.
- *Optimal:* oracle with perfect prediction of the User Behavior Predictor

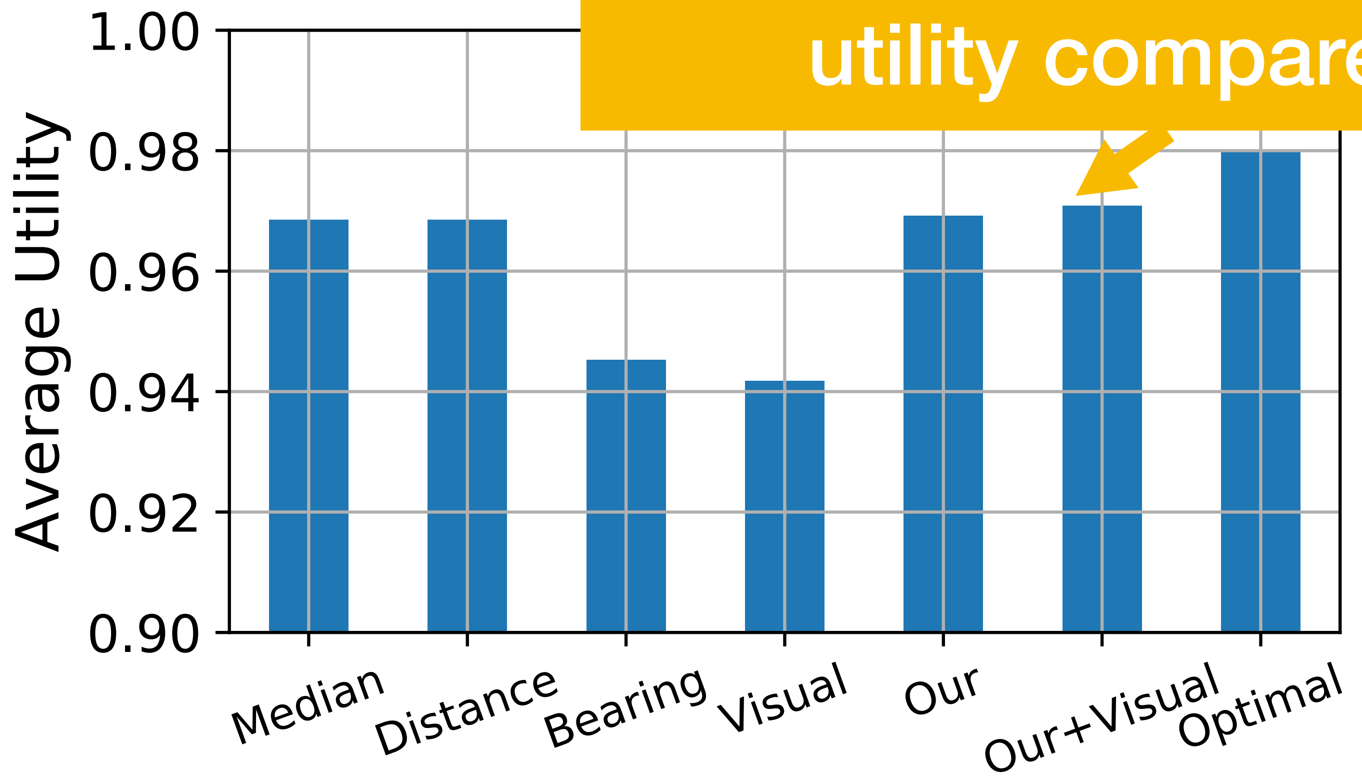
[1] NARAYANAN, A., RAMADAN, E., MEHTA, R., HU, X., LIU, Q., FEZEU, R. A. K., DAYALAN, U. K., VERMA, S., JI, P., LI, T., QIAN, F., AND ZHANG, Z.-L. Lumos5g: Mapping and predicting commercial mmwave 5g throughput, IMC '20

UC RIVERSIDE

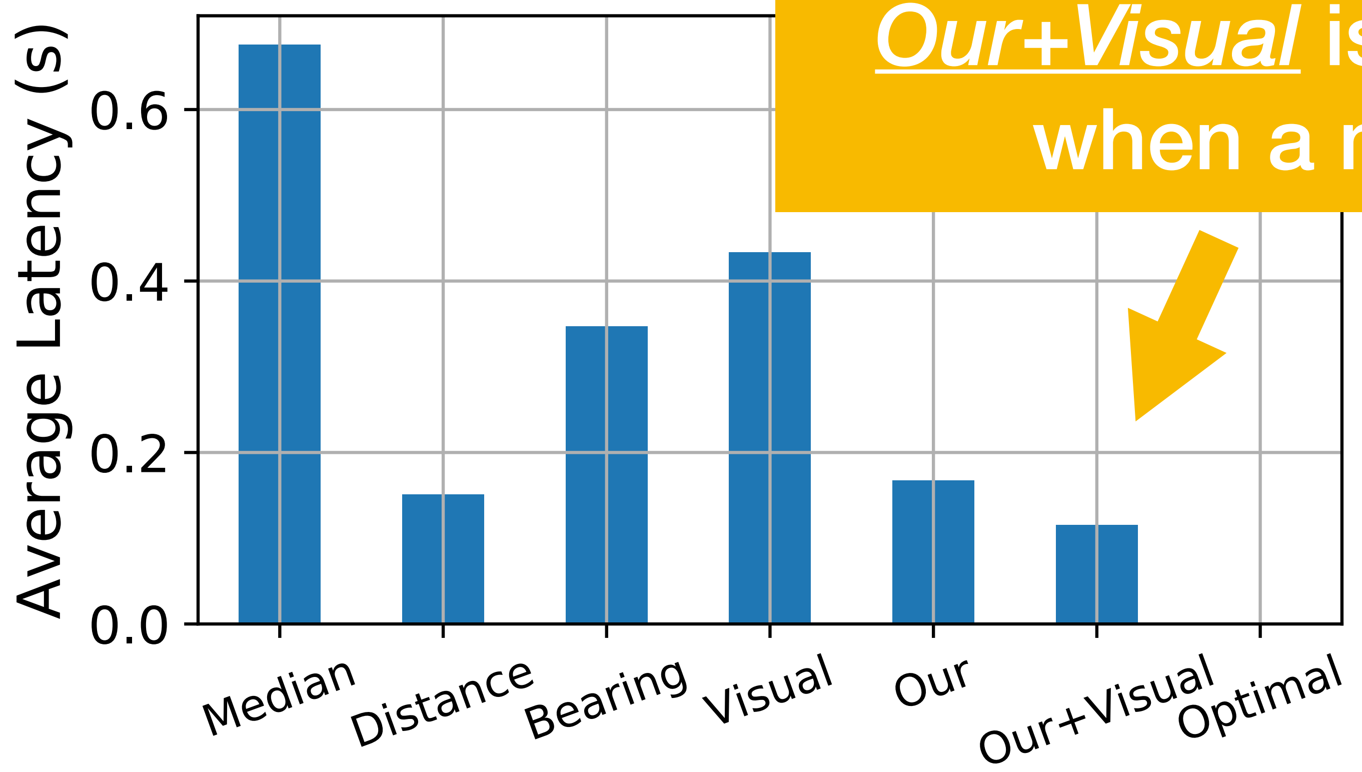
Evaluation of the End-to-end Simulation



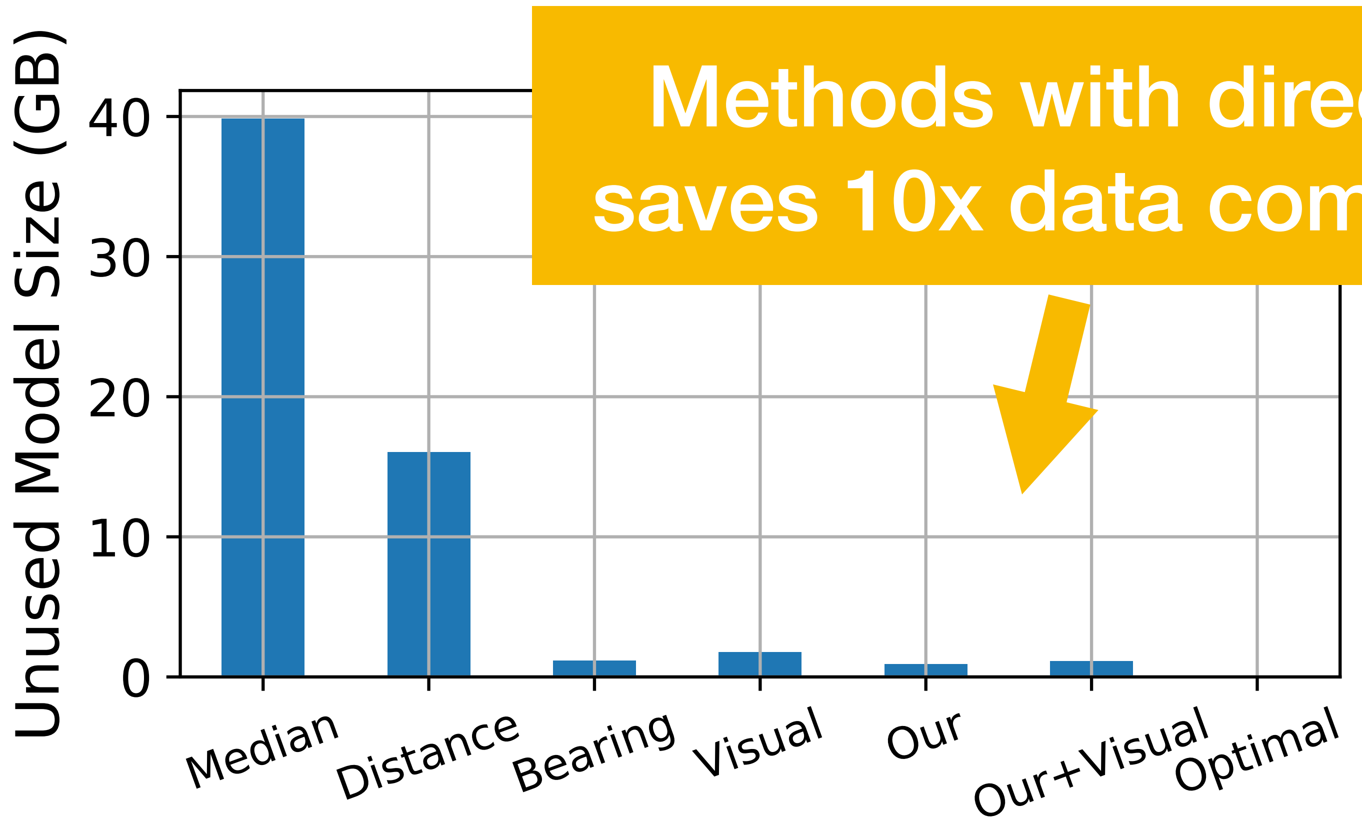
Our+Visual misses the fewest 3D models



Our+Visual can maintain comparable utility compared to Optimal



Our+Visual is the most responsive when a miss event occurs



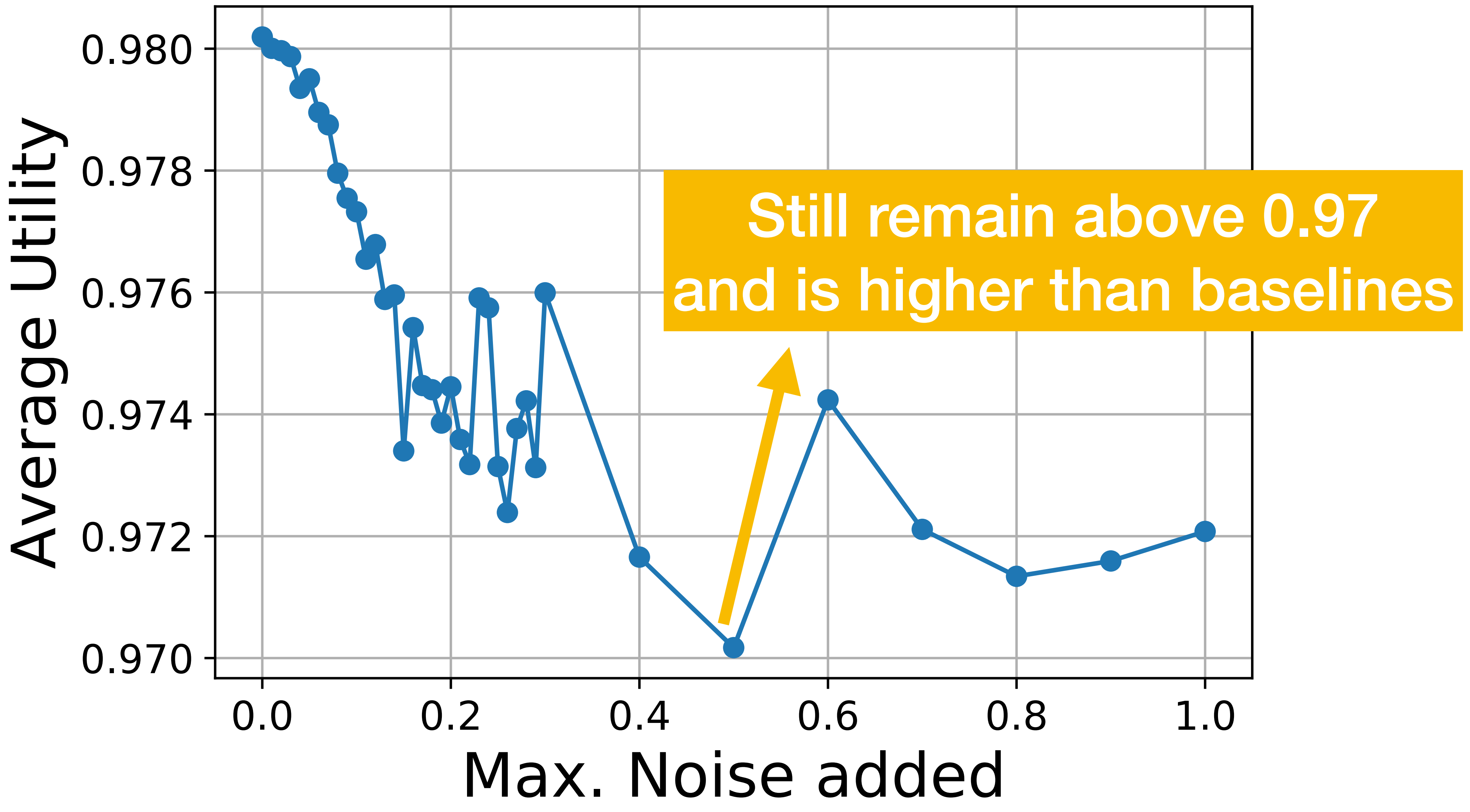
Methods with direction information saves 10x data compared to *Distance*

Our+Visual has the most balanced performance on all metrics

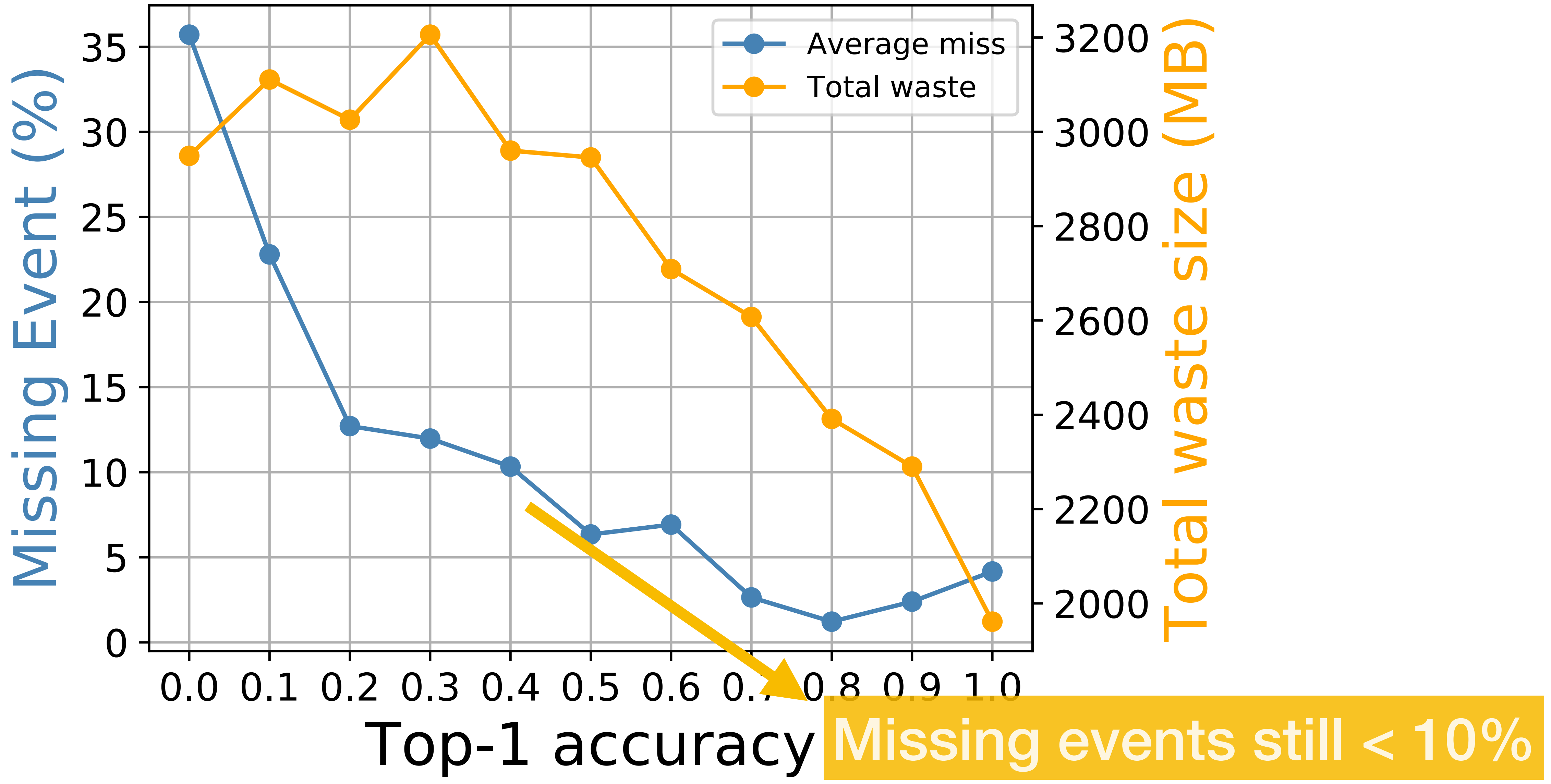
UC RIVERSIDE

Robustness to noisy predictions

- Examine the impact of noisy prediction on two predictor modules:
  - When testing 1 module, we assume perfect prediction on the other to isolate the impact
    - <u>3D Model Characterizer:</u>



    - <u>User Behavior Predictor:</u>



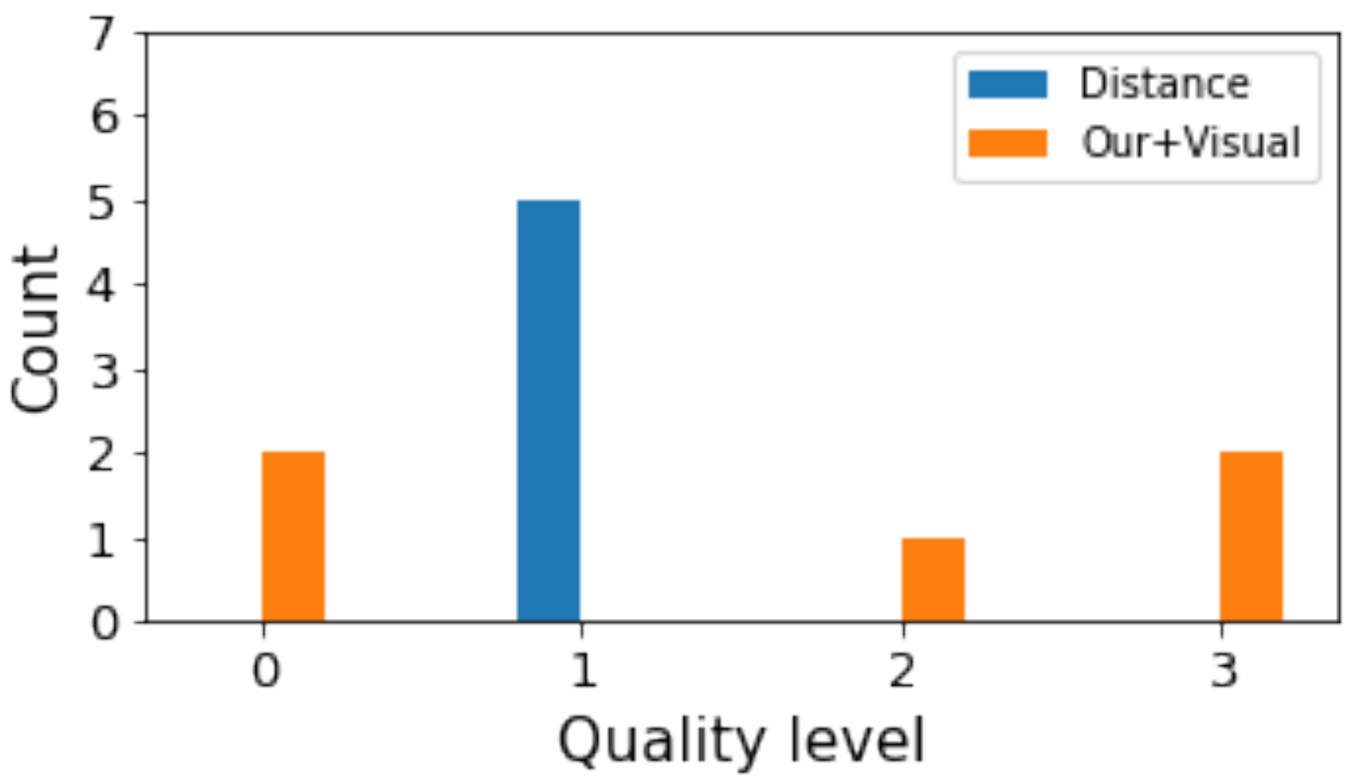
Our framework is robust even with poor performance of individual modules

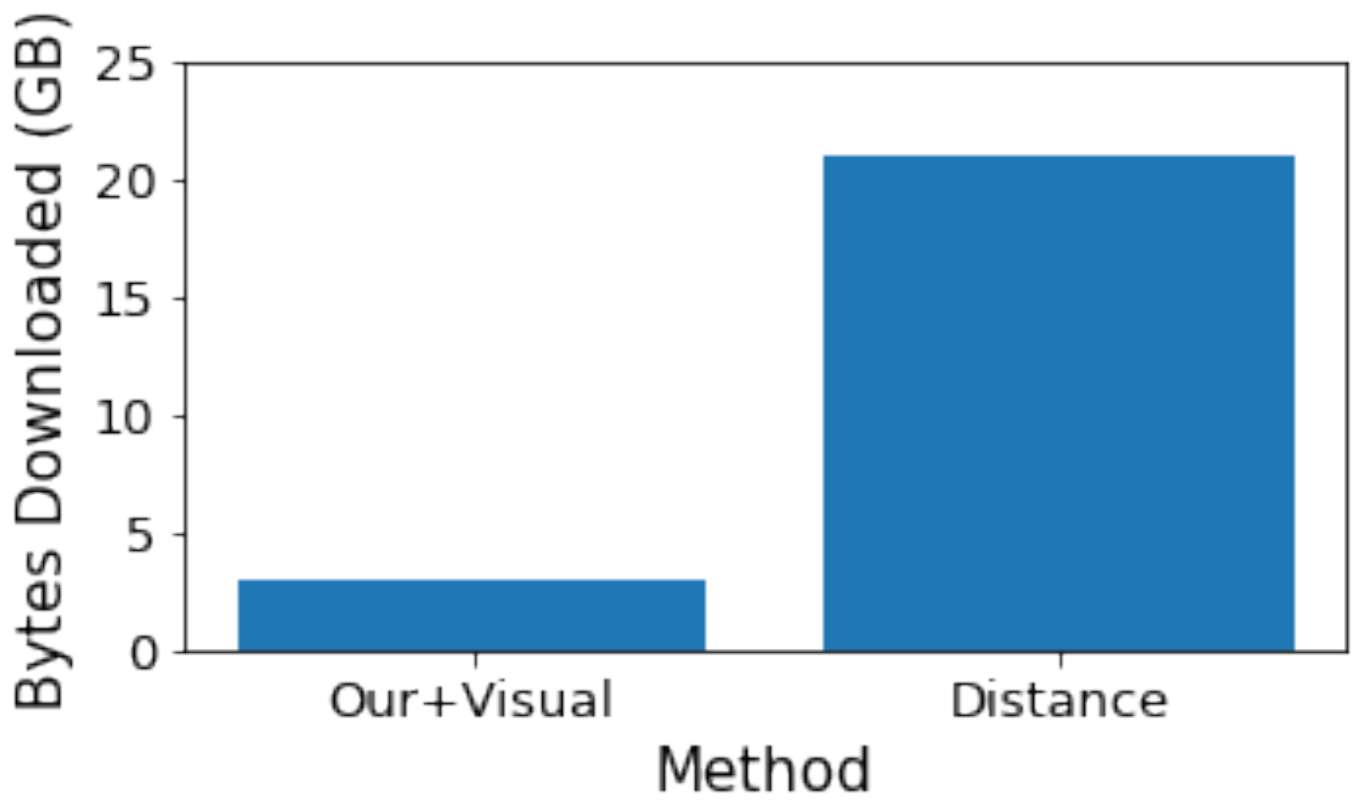
UC RIVERSIDE

Prototype Application

- A prototype Unity AR application that request 3D models that pre-determined by 3D Model Scheduler from server

- We followed the similar trajectory of previous simulation trace





Our+Visual retrieves higher quality models on average while significantly saves data compared to ***Distance*** baseline

UC RIVERSIDE

Conclusion

World-scale AR applications require user to download 3D models on-the-fly due to large geographical scale

Our framework optimizes which 3D models to download and when, by characterizing 3D model quality-latency tradeoffs and predicting AR user behavior utilizing AR visual data

Our framework misses the fewest 3D models and maintains high utility without wasting network bandwidth, compared to baselines

UC RIVERSIDE

Thank you
Questions?

UC RIVERSIDE